

**IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION**

IMPLICIT, LLC	Plaintiff,	Case No. 2:19-cv-00040-JRG-RSP (Lead Case)
vs.		
IMPERVA, INC.,	Defendant.	Case No. 2:19-cv-00039-JRG-RSP
vs.		
FORTINET, INC.,	Defendant.	Case No. 2:19-cv-00037-JRG-RSP
vs.		
JUNIPER NETWORKS, INC.	Defendant	

DEFENDANTS' JOINT RESPONSIVE MARKMAN BRIEF

TABLE OF CONTENTS

	<u>Page</u>
I. Introduction.....	1
II. Background.....	2
A. Asserted Patents	2
B. Prior Claim Construction Proceedings.....	3
III. Agreed Constructions.....	6
IV. Implicit is Collaterally Estopped from Arguing Constructions that Differ from the Court’s Constructions in <i>NetScout</i>	6
V. “Sequence of Routines” and “One or More Routines” Limitations	9
A. Implicit’s Statements in the Prosecution History.....	10
B. Implicit’s Effort to Ignore the Prosecution History Fails	13
VI. “List of Conversion Routines”.....	17
VII. Terms Regarding “Execute a Transmission Control Protocol to Convert [One or More] [At Least One of the] Packets Having a TCP Form Into a Different Format”	17
A. Collateral Estoppel.....	17
B. “Convert One or More Packets Having a TCP Format Into a Different Format” and similar terms	20
1. Outermost Header	21
2. Pointer Defining Outermost Header	22
3. Moving a Reference.....	23
C. “[Execute] [Perform] a Transmission Control Protocol” and Related Terms	25
VIII. “[S]ession Associated With A [Transport Layer/Different] Protocol” and similar terms.....	27
IX. Conclusion	29

TABLE OF AUTHORITIES

	Page(s)
CASES	
<i>Allergan Sales, LLC v. Sandoz Inc.</i> , No. 2:12-cv-207-JRG, 2016 U.S. Dist. LEXIS 41013 (E.D. Tex. Mar. 29, 2016)	9
<i>Alloc, Inc. v. Int’l Trade Comm’n</i> , 342 F.3d 1361 (Fed. Cir. 2003).....	21
<i>DietGoal Innovations LLC v. Chipotle Mexican Grill, Inc.</i> , 70 F. Supp. 3d 808 (E.D. Tex. 2014) (Bryson, J.)	7, 8
<i>Epcon Gas Sys. Inc. v. Bauer Compressors, Inc.</i> , 279 F.3d 1022 (Fed. Cir. 2002).....	7
<i>Implicit LLC v. Sandvine Corp.</i> , No. 2:18-cv-00054-JRG, Dkt. 17 (E.D. Tex. Dec. 17, 2019)	1, 8
<i>Implicit Networks, Inc. v. F5 Networks, Inc.</i> , No. 3:10-CV-3365, Dkt. No. 93, 2012 WL 669861 (N.D. Cal. Feb. 29, 2012)	3, 6
<i>Implicit Networks, Inc. v. F5 Networks, Inc.</i> , No. 3:14-CV-2856, Dkt. No. 57, 2015 WL 2194627 (N.D. Cal. May 6, 2015).....	3, 4, 5, 22
<i>Implicit, LLC v. Huawei Technologies USA, Inc., et al.</i> , 6:17-CV-182, 2018 WL 1169137 (E.D. Tex. Mar. 6, 2018) (Gilstrap, J.)	4
<i>Implicit, LLC v. NetScout Sys., Inc.</i> , No. 2:18-cv-00053-JRG, Dkt. 222 (E.D. Tex. Dec. 13, 2019)	passim
<i>Implicit, LLC v. NetScout Sys., Inc.</i> , No. 2:18-cv-53-JRG, Dkt. No. 111 (E.D. Tex. Apr. 15, 2019)	4
<i>Implicit, LLC v. Trend Micro, Inc.</i> , No. 6:16-CV-80, Dkt. No. 115, 2017 WL 1190373 (E.D. Tex. Mar. 29, 2017) (Gilstrap, J.)	4
<i>In re Arunachalam</i> , 709 F. App’x 699 (Fed. Cir. 2017)	6
<i>In re Freeman</i> , 30 F.3d 1459 (Fed. Cir. 1994).....	8
<i>Nestle USA, Inc. v Steuben Foods, Inc.</i> , 884 F.3d 1350 (Fed. Cir. 2018).....	7

TABLE OF AUTHORITIES **(Continued)**

	Page(s)
<i>Ohio Willow Wood Co. v. Alps S., LLC</i> , 735 F.3d 1333 (Fed. Cir. 2013).....	6
<i>Omega Eng’g, Inc. v. Raytek Corp.</i> , 334 F.3d 1314 (Fed. Cir. 2003).....	16
<i>Southwall Techs., Inc. v. Cardinal IG Co.</i> , 54 F.3d 1570 (Fed. Cir. 1995).....	22
<i>State Farm Mut. Auto. Ins. Co. v. LogistiCare Solutions, LLC</i> , 751 F.3d 684 (5th Cir. 2014)	7
<i>Tinnus Enters., LLC v. Telebrands Corp.</i> , No. 6:15-CV-00551-RWS, 2019 U.S. Dist. LEXIS 114601 (E.D. Tex. Apr. 16, 2019)	6
<i>Vitronics Corp. v. Conceptiontronic, Inc.</i> , 90 F. 3d 1576 (Fed. Cir. 1996).....	13
 RULES	
Patent Rule 4-5(b)	1
 OTHER AUTHORITIES	
David Mosberger, “Scout: A Path-Based Operating System,” Doctoral Dissertation Submitted to the University of Arizona.....	4

Table of Exhibits

Exhibit Number	Description
1	'683 Patent File History, Preliminary Amendment dated June 6, 2013
2	'163 <i>Inter Partes</i> Reexam (Reexamination Control No. 95/000,659), Comments to October 1, 2012 ACP
3	Implicit's Patent Local Rule 4-5 Opening Claim Construction Brief in <i>Implicit Networks, Inc. v. F5 Networks, Inc.</i> , No. 3:14-CV-2856, Dkt. No. 57, 2015 WL 2194627 (N.D. Cal. May 6, 2015)
4	'163 <i>Ex Parte</i> Reexamination (Reexamination Control No. 90/010,356), "Amendment and Response to Office Action Mailed July 7, 2009"
5	Mosberger ("Scout: A Path-Based Operating System, David Mosberger, Doctoral Dissertation Submitted to the University of Arizona")

I. Introduction

Defendants Juniper Networks, Fortinet, and Imperva submit this Responsive Claim Construction Brief pursuant to Patent Rule 4-5(b). Defendants respectfully request that the Court adopt the Defendants' proposed constructions for the disputed claim terms as detailed below.

The asserted Implicit patents in this case have a long history that is critically relevant to the parties' claim construction disputes. The asserted patents (all of which are expired) are part of a single family of patents known as the "demultiplexing" family. This patent family has been subjected to repeated examination in the USPTO, including reexamination proceedings involving a couple of prior art references known as "Mosberger" and "Decasper." In an effort to overcome these references, Implicit made numerous binding admissions that narrowly characterize the scope of its alleged invention. The USPTO ultimately found that at least two of Implicit's earlier patents in this family were invalid despite these admissions.

Implicit has also asserted its "demultiplexing" patents in litigation against numerous technology companies. To date, each case to reach the merits has resulted in a loss for Implicit. For example, Implicit lost its first case against Juniper on summary judgment (non-infringement and invalidity) before Judge Illston in the Northern District of California. More recently, a jury found Implicit's patents not infringed after a full trial in *Implicit, LLC v. NetScout Sys., Inc.*, No. 2:18-cv-00053-JRG, Dkt. 222 (E.D. Tex. Dec. 13, 2019). Following this verdict, Implicit entered a stipulation of non-infringement in *Implicit LLC v. Sandvine Corp.*, No. 2:18-cv-00054-JRG, Dkt. 17 (E.D. Tex. Dec. 17, 2019). Notably, almost all of the claim constructions now proposed by Implicit were already presented to and rejected by this Court in *NetScout* and *Sandvine*.

Implicit's prior admissions and the Court's earlier rulings regarding these patents dictate the outcome of these claim construction proceedings as well. The Court should reject Implicit's proposed constructions—again—as they are inconsistent with this extensive litigation and

prosecution history. By contrast, Defendants’ proposed constructions are fundamentally based in the intrinsic record of the asserted patents.

II. Background

A. Asserted Patents

Implicit in this consolidated case currently alleges infringement of six related patents, U.S. Patent Nos. 8,694,683 (“the ’683 Patent”), 9,270,790 (“the ’790 Patent”), 9,591,104 (“the ’104 Patent”), 10,027,780 (“the ’780 Patent”), 10,033,839 (“the ’839 Patent”), and 10,225,378 (“the ’378 Patent”) (collectively, “the Asserted Patents”). *See* Dkt. No. 161. The Asserted Patents share a common specification and are all continuations of the same patent application that issued as U.S. Patent Nos. 6,629,163 (“the ’163 Patent”).¹ The ’163 Patent was the subject of both an *ex parte* Reexamination No. 90/010,356 (“’163 *ex parte* Reexam”) and *inter partes* Reexamination No. 95/000,659 (“’163 *inter partes* Reexam”). *See* Exs. 2, 4. The patents are also related to and share the same specification as U.S. Patent No. 7,711,857, which also underwent *inter partes* reexamination as Reexamination No. 95/000,660 (“’857 Reexam”).

The Asserted Patents are all titled “Method and System for Data Demultiplexing” and are generally directed to technology for computer message-exchange processing. The Asserted Patents purportedly resolve the issue of how to convert packet-based data into “many different intermediate formats” when transmitting data over a network. *See* ’683 Patent² at 1:24-26. To do so, they describe a “dynamic” approach by which information obtained from a portion or “packet” of a message is used to identify a series of software routines that convert the packets from the

¹ The ’790 Patent is a continuation of the ’683 Patent. The ’104 Patent, in turn, is a continuation of the ’790 Patent.

² As the Asserted Patents share a common specification, patent citations are provided to the ’683 Patent only unless otherwise noted.

original or “source” format to another “target” format. *See id.* at 2:4-6 (“It would be desirable to have a technique for dynamically identifying a series of conversion routines for processing data.”); 2:44-53 (“When a packet of a message is received, the conversion system in one embodiment searches for and identifies a sequence of conversion routines . . . used to convert the message from the source format to the target format using various intermediate formats.”). Once determined, the system stores the identified sequence which is used when the next packet of the message arrives; subsequent packages of the same message are processed according to the stored sequence. *Id.* at 2:55-61.

The Asserted Patents distinguish their dynamic approach from the static-nature of the prior art. *See id.* at 1:45-2:3. Specifically, prior-art “computer systems typically use predefined configuration information to load the correct combination of conversion routines for processing data.” *Id.* at 1:48-50. However, data to be converted “may take many different formats that may not be known until the data is received” and therefore “[t]he overhead of statically providing each possible series of conversion routines is very high.” *Id.* at 1:54-59. During the ’163 *inter partes* Reexam, the patentee explained that “the invention requires the sequence of conversion routines (that form the paths) to be identified at run-time, and disavows prior art systems (like Mosberger) that use pre-configured paths, which are defined at ‘build-time’ before the first packet of a message is received.” Ex. 4 at 18.

B. Prior Claim Construction Proceedings

This Court is familiar with the Asserted Patents; it has previously construed terms from the Asserted Patents in three different cases in the last four years.³ The Court first addressed the

³ The Northern District of California has also issued claim construction orders for the ’163 Patent in *Implicit Networks, Inc. v. F5 Networks, Inc.*, No. 3:10-CV-3365, Dkt. No. 93, 2012 WL 669861 (N.D. Cal. Feb. 29, 2012) (“*F5 Networks I*”) and the ’683 Patent in *Implicit Networks, Inc. v. F5*

demultiplexing patents in *Implicit, LLC v. Trend Micro, Inc.*, No. 6:16-CV-80, Dkt. No. 115, 2017 WL 1190373 (E.D. Tex. Mar. 29, 2017) (Gilstrap, J.) (“*Trend Micro*”) when it construed terms from both the ’683 and ’790 Patents. A year later, the Court construed terms from the ’683, ’790, and ’104 Patents in *Implicit, LLC v. Huawei Technologies USA, Inc., et al.*, 6:17-CV-182, 2018 WL 1169137 (E.D. Tex. Mar. 6, 2018) (Gilstrap, J.) (“*PAN*,” an acronym for Palo Alto Networks, Inc., the only remaining defendant in the *Huawei* case at the time of the claim construction hearing). Most recently, this Court again construed the ’683, ’790, and ’104 Patents in *Implicit, LLC v. NetScout Systems, Inc.*, No. 2:18-cv-53-JRG, Dkt. No. 111 (E.D. Tex. Apr. 15, 2019) (“*NetScout*”).

In *NetScout*, the Court considered many of the same terms at issue here. In construing “sequence of [two or more] routines” and “list of conversion routines,” the Court followed the prior *F5 Networks II*, *Trend Micro*, and *PAN* decisions in finding patentee’s statements regarding the Mosberger prior art reference⁴ during the ’163 *ex parte* Reexam amounted to a disclaimer of claim scope. *NetScout* at 10-14. It refined the prior constructions, noting that “[t]he patentee explained that the claimed invention uses software routine *arrangements* that *were not created prior* to receiving a first packet of the message,” and construed the terms as “an ordered arrangement of [two or more] software [conversion] routines that was not selected from a set of arrangements created before receiving a first packet of the message.” *Id.* at 14 (emphasis in original).

Networks, Inc., No. 3:14-CV-2856, Dkt. No. 57, 2015 WL 2194627 (N.D. Cal. May 6, 2015) (“*F5 Networks II*”).

⁴ David Mosberger, “Scout: A Path-Based Operating System,” Doctoral Dissertation Submitted to the University of Arizona.

“State information” was also construed in *NetScout* to mean “information that is specific to a software routine for a specific message, that can be used for all packets of the message, and that is not information related to an overall path.” The parties have agreed to this construction in this case. *See* Dkt. No. 161 at 2.

Regarding “convert one or more packets . . . into a different format” and related terms, the Court relied on patentee’s statements during the ’163 and ’857 *inter partes* Reexams as well as Implicit’s statements in *F5 Networks II* to hold that the format of a packet is defined by its outermost header. *See NetScout* at 25-29. Specifically, during the ’163 *inter partes* Reexam, patentee stated, “[w]hether a packet is an IP format is determined by the structure of its header” (Ex. 2 at 12) and “[t]he format of the IP packet is not changed unless the actual structure of the header itself is changed from that shown in Figure 2 to a different type of header (*e.g.*, a TCP header)” (*id.* at 24). *See NetScout* at 25-26. In connection with the ’857 Reexam, the patentee submitted expert testimony stating, “only the structure of the outermost header determines whether the packet is an IPv4 packet or whether it employs some other protocol.” Dkt. 208-9 (Ng Decl.) at ¶ 6; *NetScout* at 26. The Court also found it “noteworthy” that Implicit “understood that the relevant ‘format’ of a packet is determined by its outermost header,” based on Implicit’s argument that “[e]ach message packet can include different layers in different data formats” and, for example, “software routines associated with an Ethernet protocol will process the packet first, as the outermost layer of the packet is in an Ethernet format” in *F5 Networks II*. *NetScout* at 26 (citing Ex. 3 at 4-5). The Court also expressly rejected Implicit’s argument that “convert” could encompass “merely moving a reference” as “lacking support in the record” and contradicted by evidence in the ’683 Patent’s prosecution history that “implies that ‘converting’ requires modifying headers.” *NetScout* at 27 (citing Ex. 1 at 19).

For “execute a [Transmission Control Protocol (TCP) / different protocol]” and related terms, the Court held it presented substantially the same issue relating to the “outermost header” as the “convert” terms and construed the terms to mean “operate on one of more packets whose outermost header is a [TCP / different protocol] header.” *Id.* at 33-36.

The Court’s constructions in *NetScout* are currently on appeal before the Federal Circuit in *Implicit, LLC v. Sandvine Corporation*, CAFC No. 20-1362. Implicit is also challenging these constructions in post-trial briefing in *NetScout*.

III. Agreed Constructions

Defendants agree that the Parties have reached agreement as to the constructions of “message,” “state information,” “the packet of the message,” and “key [value].” Dkt. 208 at 6-7.

IV. Implicit is Collaterally Estopped from Arguing Constructions that Differ from the Court’s Constructions in *NetScout*

Despite the Court’s claim construction ruling in *NetScout* (Dkt. 111), Implicit continues to argue for different constructions of those claim terms in this action. This is improper because Implicit is estopped from offering different constructions that it already presented and were rejected in *NetScout*. See *In re Arunachalam*, 709 F. App’x 699, 703 (Fed. Cir. 2017) (patentee collaterally estopped from construing claims because the same terms were previously construed).

“In patent cases, issue preclusion is analyzed under the law of the regional circuit.” *Tinnus Enters., LLC v. Telebrands Corp.*, No. 6:15-CV-00551-RWS, 2019 U.S. Dist. LEXIS 114601, at *8 (E.D. Tex. Apr. 16, 2019) (citing *United Access Techs., LLC v. Centurytel Broadband Servs. LLC*, 778 F.3d 1327, 1330 n.1 (Fed. Cir. 2015)). However, the Federal Circuit “appl[ies its] . . . own precedent to those aspects of such a determination that involve substantive issues of patent law.” *Ohio Willow Wood Co. v. Alps S., LLC*, 735 F.3d 1333, 1342 (Fed. Cir. 2013) (citing *Aspex*

Eyewear, Inc. v. Zenni Optical Inc., 713 F.3d 1377, 1380 (Fed. Cir. 2013)). The Fifth Circuit has held that collateral estoppel (also known as issue preclusion) applies if:

First, the issue under consideration in a subsequent action must be identical to the issue litigated in a prior action. Second, the issue must have been fully and vigorously litigated in the prior action. Third, the issue must have been necessary to support the judgment in the prior case. Fourth, there must be no special circumstance that would render preclusion inappropriate or unfair.

State Farm Mut. Auto. Ins. Co. v. LogistiCare Solutions, LLC, 751 F.3d 684, 689 (5th Cir. 2014) (citing *United States v. Shanbaum*, 10 F.3d 305, 311 (5th Cir. 1994)). “Importantly, the collateral estoppel effect of a prior district court decision is not affected by the fact that an appeal has been taken from the decision.” *DietGoal Innovations LLC v. Chipotle Mexican Grill, Inc.*, 70 F. Supp. 3d 808, 812 (E.D. Tex. 2014) (Bryson, J.).

All of these factors weigh in favor of the application of collateral estoppel.

First, the disputed terms are substantially identical to the terms previously construed. Many of the disputed terms are exactly the same and others (like “one or more routines”) use slightly different wording but present the same issues. The lack of complete word-for-word identity between some of the terms does not preclude application of collateral estoppel. *See Nestle USA, Inc. v. Steuben Foods, Inc.*, 884 F.3d 1350, 1352 (Fed. Cir. 2018) (“[O]ur precedent makes clear that collateral estoppel is not limited to patent claims that are identical. Rather, it is the identity of the *issues* that were litigated that determines whether collateral estoppel should apply.” (quotation marks omitted)). Nor does it matter that additional patents are at issue in this case given their relationship to the patents in *NetScout*. *See Epcon Gas Sys. Inc. v. Bauer Compressors, Inc.*, 279 F.3d 1022, 1030 (Fed. Cir. 2002) (“the same term or phrase should be interpreted consistently where it appears in claims of common ancestry”).

Second, these terms were “fully and vigorously litigated in the prior action,” culminating in final judgments. “The requirement that the issue have been actually decided is generally

satisfied if the parties to the original action disputed the issue and the trier of fact decided it.” *In re Freeman*, 30 F.3d 1459, 1466 (Fed. Cir. 1994). Here, the parties in *NetScout* disputed the claim terms, and the Court ruled on those disputes—this issue was fully and vigorously litigated. Accordingly, the Court need not consider “whether the prior finding . . . was correct” in determining whether collateral estoppel applies. *Id.* at 815. Implicit’s pending appeal does not negate this factor. *DietGoal*, 70 F. Supp. 3d at 812.

Third, construction of the disputed terms was necessary to support the final judgments in the consolidated Sandvine and NetScout Actions. The Sandvine Stipulation of non-infringement did not involve a trial at all; it lists the constructions and states, “Implicit and Sandvine stipulate that under the Court’s construction of the above terms, there is no infringement of the Asserted Claims by the Accused Products.” *Implicit, LLC v. Sandvine Corporation*, No. 2:18-cv-00054-JRG (“*Sandvine*”), Dkt. 17 at 4 (E.D. Tex. Dec. 17, 2019). Thus, the Sandvine Stipulation explicitly sets forth the collective basis for the final judgment of non-infringement—the claim terms and associated constructions identified therein. The Sandvine Final Judgment confirmed the judgment of noninfringement was “pursuant to the Stipulation” that was “based on the Court’s claim construction order.” *Sandvine*, Dkt. 17 at 4; *id.*, Dkt. 18 at 1. Nothing suggests any term or its construction is an alternative basis for relief, *and importantly* the stipulation does not include all of the claim terms construed by the Court. Indeed, if the terms in the Sandvine Stipulation were not necessary to the final judgment, the parties would be seeking an improper advisory opinion from the Federal Circuit.

Moreover, consideration of this factor “does not mean that the finding must be so crucial that, without it, the judgment could not stand. Rather, the purpose of the requirement is to prevent the incidental or collateral determination of a nonessential issue from precluding reconsideration

of that issue in later litigation.” *Allergan Sales, LLC v. Sandoz Inc.*, No. 2:12-cv-207-JRG, 2016 U.S. Dist. LEXIS 41013, at *22-23 (E.D. Tex. Mar. 29, 2016). As this Court has stated, “[c]laim construction is not incidental or non-essential to issues of validity and infringement. Indeed, it is the first step in the process of deciding those issues.” *Id.* Thus, the construction of these terms was essential to the final judgment of non-infringement in *NetScout*, which must have been based on Court’s claim construction order as well. The Court should therefore find that the construction of these terms was “at least implicitly essential to the judgment[s]” of non-infringement in the Sandvine and NetScout Actions. *Id.*

Finally, there is “no special circumstance that would render preclusion inappropriate or unfair” here. In fact, it would be unfair if Implicit were not precluded, as it would be able to relitigate the same constructions it previously lost in the *NetScout*. The terms for which Implicit is collaterally estopped from arguing new constructions are addressed within the relevant sections below.

V. “Sequence of Routines” and “One or More Routines” Limitations

Defendants propose the following construction for the limitations “sequence of [two or more] routines” (’683 Patent, claims 1, 4, 5, 8, 9, 24; ’790 Patent, claims 1, 3, 4, 5, 8, 10, 12, 15, 17, 18; ’104 Patent, claims 1, 3, 5, 10, 12, 13, 16), “one or more routines” (’780 Patent, claims 1, 2, 3, 16, 20; ’839 Patent, claim 1; ’378 Patent, claims 1, 2, 3, 16, 17, 20), and “routines in the sequence of routines” (’683 Patent, claim 8).⁵

⁵ Should the Court not entirely agree with either party’s proposal regarding these limitations, Defendants propose in the alternative that the Court construe these limitations as it did in *NetScout*.

Implicit's Proposed Construction	Defendants' Proposed Construction
"an ordered arrangement of [two or more] routines that was not identified (i.e., configured) prior to receiving a first packet of a message"	"[two or more] software routines arranged in a sequence that was not established in a chain of modules connected before receiving a first packet of the message"

The parties agree that these terms are related to the claimed "path" of Implicit's alleged invention, including "how the software 'creates' that 'path.'" Dkt. 208 at 7. However, that process is much more clearly articulated in Defendants' proposed construction, which tracks the representations made by Implicit to the Patent Office. Defendants' proposed construction is also similar to the one adopted by the Court in *NetScout*, with a few clarifications in an attempt to address some of the legally improper (and ultimately unsuccessful) arguments that Implicit attempted during trial in that case.

A. Implicit's Statements in the Prosecution History

The language of Defendants' proposed construction comes directly from the intrinsic record, namely the prosecution history for the '683 Patent. During that prosecution, Implicit argued that the operation of its invention could be distinguished from how the prior art "Scout" system in the Mosberger prior art "create[s] a set of paths comprising various sequences of modules." Ex. 1 at 11.

By way of background, Mosberger provides examples of data flow graphs that show the available routines (modules) that could be applied to a packet during processing. *See* Ex. 5. Figure 2.4 of Mosberger depicts an example of a path through these routines:

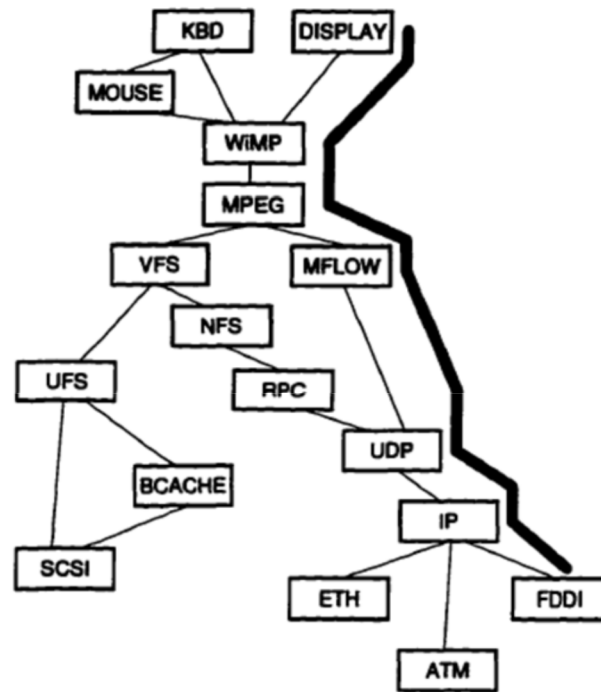


Figure 2.4: Example Path in Modular System

Id. at 38. Mosberger describes the routing of a particular packet through Figure 2.4:

With this approach, IP itself can make the decision whether or not the path should extend to ATM or FDDI. . . .

Path creation is then forwarded to that next module. This process repeats itself until there is no next module . . .

Id. at 40-41. In distinguishing Mosberger, Implicit noted that its patent specification (the same specification in the Asserted Patents) “clearly states that the invention requires the sequence of conversion routines (that form the paths) to be identified at run-time, and disavows prior art systems (like Mosberger) that use **pre-configured** paths, which are defined at ‘build-time’ before the first packet of a message is received.” Ex. 4 at 18. More specifically, Implicit explained that the graphs in Mosberger are “configured” before the first packet arrives: “once the module graph is defined by the developer, the system, including all of its paths, are “carved in stone” (i.e., configured and unalterable).” *Id.* at 12. Of critical importance, Implicit represented that these

paths were “configured” before the arrival of the first packet, even though the decision about how to process the first packet would be made at each node, iteratively, based on the content of the packet. On this point, Implicit focused on Figure 3.6 of Mosberger (right), and Implicit included an image of it in the prosecution history. *Id.*

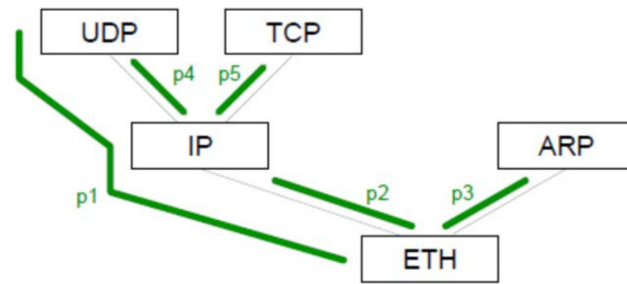


Figure 3.6: Paths Versus Classifiers

at 14. Implicit’s explanation of this Figure in Mosberger was as follows:

As seen in Figure 3.6 above (cited by the Examiner), there are a number of pathways (p1-p5) for the data to travel, but the selection of modules for each path is configured before receipt of the message packets. . . . In this example, the ETH module selects the initial pathway (i.e., pathway p1, p2 or p3) based on the protocol header in the first packet of the message. (Mosberger pages 88-89). If the protocol header is completely intact and suggests that packet is UDP, then the packet classification process will result in the selection of pre-configured path 1. If, however, the packet has been fragmented by IP and does not include higher level headers, then the classification process will result in selection of pre-configured path 2. (Mosberger, page 87). At that point, after the packet has been reassembled, the IP module’s packet classifier will determine whether the packet is a UDP or a TCP packet and route that packet accordingly (i.e., down pre-configured path 4 or pre-configured path 5). (Mosberger, page 87). Importantly, however, those sequences of IP-UDP via pathway p4 and IP-TCP via pathway p5 have also been identified prior to run-time. Again, the only question resolved at run time is which preconfigured path – with predetermined sequences of modules – should be selected.

Id. at 14. Unlike the Asserted Patents, Implicit explained that in Mosberger, “all interconnectivity is predetermined and compiled into the operating system before it ever receives a single packet of data.” *Id.* at 22. Implicit argued that “the sequence of conversion routines (or ‘path’) is not configured prior to receiving the first packet of a message.” *Id.* at 18. Likewise, Implicit confirmed that “a routing decision in Mosberger is merely a selection between multiple predefined pathways.” *Id.* at 28. Thus, Implicit disclaimed iterative navigation through modules in a data graph, where the available modules at each decision point are preset before the first packet arrived.

The question, however, is how to best capture the essence of that disclaimer for the jury. In this regard, Implicit described Mosberger’s function for creating paths as something that “*establishes the various chains* [of modules] through the path structure.” Ex. 1 at 11 (quoting Ex. 5 at 81). It was on this basis that Implicit represented to the USPTO that Mosberger “does not teach or suggest” the concept of a “sequence of routines.” *Id.* Implicit further told the USPTO that “[t]he knowledge of which modules to *connect* together is compiled into the Scout kernel *at compile time*”—that is, *before* receiving the first packet of the message. *Id.* These binding admissions make clear that a “sequence of routines” necessarily excludes routines arranged in a sequence that was already previously “established in a chain of modules connected before receiving a first packet of the message.”

B. Implicit’s Effort to Ignore the Prosecution History Fails

Implicit’s primary argument is that Defendants’ construction is driven by a non-infringement argument that prevailed in *NetScout* and which Implicit believes is “improper.” *See* Dkt. 208 at 13. That is not an appropriate claim construction argument. To the contrary, the Federal Circuit has made clear that claim construction is an independent first step that precedes any analysis of infringement. *See, e.g., Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1581-82 (Fed. Cir. 1996) (“A literal patent infringement analysis involves two steps: [1] the proper construction of the asserted claim and [2] a determination as to whether the accused method or product infringes the asserted claim as properly construed.”).⁶ Implicit further argues that Defendants’ proposed construction would “exclude the embodiments of the Implicit Patents” (Dkt.

⁶ In any event, Implicit’s argument is incorrect as Defendants’ non-infringement arguments would apply equally even if the Court were to adopt Implicit’s proposed construction or the construction adopted in *NetScout*. Indeed, in the event that the Court elects not to adopt Defendants’ proposed construction, the construction adopted in *NetScout* would present an acceptable alternative.

208 at 13), but never explains what embodiments it is referring to, much less identify any alleged inconsistency. To the contrary, as demonstrated above, Defendants' proposed construction simply incorporates the very language that *Implicit* used to distinguish its supposed invention from prior art in the intrinsic record.

Implicit's arguments in favor of its alternative proposed construction likewise fail. As an initial matter, *Implicit*'s proposal is the same one as was considered and rejected by this Court in *NetScout*. *Implicit* even concedes that the Court's prior ruling on this term formed the basis for *NetScout*'s successful jury verdict of non-infringement. Thus, having already presented its construction once before and lost on the merits, *Implicit* is estopped from raising it again. *See* Section IV above.

Implicit's arguments in favor of its construction also fail on their merits. For example, *Implicit* argues that its proposed construction "captures [a] basic concept in the claims that [a] 'path,' which is a data structure, is created after the first packet arrives" Dkt. 208 at 7. But the claims directed to this alleged "basic concept" are clear when "paths" are created. For example, claim 1 of the '683 Patent states that a "path" is created "based on an identification of information in a received packet of a message," while claim 1 of the '790 Patent states that a path "is identified based on a key located in one of the received packets."

Implicit's argument regarding the motivation of its proposed construction is at best a red herring, as no Defendant has ever taken the position that *Implicit*'s claimed "paths" are (or could be) created before any packet of a message arrives. *Implicit*'s proposed construction, moreover, does not shed light on when (or how) any "path" is created. Instead, *Implicit*'s proposed construction seems directed to generating ambiguity about and offers no guidance for what it

means for a sequence of routines to be “identified, (i.e. configured) prior to receiving a packet of a message.”

Implicit’s proposal is also an improper attempt to recover broad claim scope that it expressly surrendered during prosecution. Under Implicit’s construction, *any* selection of routines from a predetermined set of sequences of routines would fall within the scope of Implicit’s claims, at least whenever at least one data structure making reference to the selection is created *after* the receipt of a packet of a message. However, Implicit’s present position is at odds with its position during prosecution. For example, during the prosecution of the ’683 Patent, Implicit argued that the Mosberger reference did not teach the creation of a path “based on identification of information in a packet of a message,” because Mosberger only “describe[d] how to *select* [an] appropriate path from amongst [a] finite set of previously-created paths” based upon the contents of a message.” Ex. 1 at 8-11 (emphasis added). Indeed, Implicit expressly distinguished Mosberger on the basis that “importantly,” the “set of paths [compiled into Mosberger at compile time] [wa]s *finite*,” with Mosberger not teaching “creation of new paths after initialization” of its software. *Id.* at 11 (emphasis added). Implicit also explicitly distinguished the sequences in its claimed “paths” from the prior art on the basis that the ordering of the sequences in its invention was not determined until packets arrive. *See, e.g., id.* at 12 (arguing that Mosberger does not teach the creation of a path that includes a sequence of routines because it “teaches that when a message is received, a path is selected . . . from a set of possible paths[] which were *created before the message was received*”) (some emphasis added). Implicit’s statements during prosecution thus disavow the use of predetermined sequences of “routines” whose order has been established prior to the receipt of a packet in any message.

The distinction raised by Implicit during prosecution—between *selecting* among predetermined sequences of routines, on the one hand, and *creating* a sequence of routines in response to analysis of a packet, on the other—is the basis of Defendants’ proposed construction, which assists the jury by explaining that the claimed sequences of “routines” must be “arranged in a sequence that was *not* established . . . *before* receiving a first packet of the message” (emphasis added).

Finally, Implicit argues that the claim term “one or more routines” should be interpreted differently from the similar terms found in other claims. *See* Dkt. 208 at 14-16. This argument is also incorrect. It is axiomatic that similar claims from related patents should be construed in the same way. *See Omega Eng’g, Inc. v. Raytek Corp.*, 334 F.3d 1314, 1334 (Fed. Cir. 2003).

Nor does Implicit’s argument find any support in the passage that it cites from the prosecution history for the ’780 Patent. *See* Dkt. 208-3 at 8-9. That portion of the prosecution history simply incorporates the same arguments made in connection with earlier patents in the demultiplexing family, including the ’104 and ’683 Patents. *See id.* Thus, if anything, this history only further confirms that these terms should be construed in the same way. Certainly, this Court did not find anything in the prosecution histories of the ’104 or ’683 Patents that would lead to different result in *NetScout*. In fact, the ’780 Patent expressly cites and incorporates by reference the passages cited above regarding Mosberger from the ’683 Patent prosecution history:

Applicant submits that the present claims distinguish over the references cited against ancestor applications for at least the reasons given in the prosecution history of U.S. Patent No. 8,694,683. *See* ’683 Prosecution History, Preliminary Amendment of June 6, 2018, at 9-20 (*distinguishing Mosberger* and Decasper). ***Those remarks are incorporated by reference herein.***

Id. at 8. Thus, Implicit’s evidence demonstrates that the same characterizations and limitations based on Mosberger described in the ’683 Patent prosecution are equally applicable for the ’780 Patent and other later patents in the same family. Accordingly, the Court should adopt the

Defendants’ proposed construction, which specifically relies on those key admissions in the intrinsic record as demonstrated above.

VI. “List of Conversion Routines”

The parties’ proposals for the term “list of conversion routines” limitation (’683 Patent claim 10) present the same issue as the “routines” issues discussed above. Resolution of the prior limitations should resolve this dispute as well, and Defendants incorporate these arguments by reference.

VII. Terms Regarding “Execute a Transmission Control Protocol to Convert [One or More] [At Least One of the] Packets Having a TCP Form Into a Different Format”⁷

A. Collateral Estoppel

These claim terms were construed by the Court in *NetScout*, and for the reasons in Section IV, Implicit is collaterally estopped from arguing for different constructions or characterizations of those constructions. Dkt. 208 at 17-27. These terms are listed in the chart below.

#	Plaintiff’s Claim Term	Plaintiff’s Proposed Construction	Sandvine Court Construction and Stipulation
5	<p>“a routine that is used to execute a Transmission Control Protocol (TCP) to convert one or more packets having a TCP format into a different format”</p> <p>“a routine that is executable to perform a Transmission Control Protocol (TCP) to convert at least one of the packets of the message into a different format”</p>	<p>Plain and ordinary meaning. No construction necessary.</p> <p>Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”</p>	<p>a routine that is used to “operate on one or more packets whose outermost header is a TCP header”/ “operable on one or more packets whose outermost header is a TCP header”</p> <p>to “convert the outermost header structure of the packet(s) from TCP to another type of header structure”</p>

⁷ This section addresses the relevant portions of the “execute” and “convert” claim terms discussed by Implicit at Dkt. 208, pages 17-27.

#	Plaintiff's Claim Term	Plaintiff's Proposed Construction	Sandvine Court Construction and Stipulation
	<p>“a routine that is used to execute a Transmission Control Protocol (TCP) to convert packets having a TCP format into a different format”</p> <p>“a particular routine that is used to execute a Transmission Control Protocol (TCP) to convert packets having a TCP format into a different format”</p> <p>'683 Patent: Claim 1 '790 Patent: Claims 1, 8, 15 '780 Patent: Claims 1, 16 '839 Patent: Claim 1 '378 Patent: Claims 1, 16</p>		<p>The Court also concluded the following: “[t]o whatever extent Plaintiff contends that the terms ‘convert one or more packets having a TCP format into a different format,’ ‘convert one or more packets in a transport layer format into a different format,’ and ‘convert packets of the different format into another format’ encompass merely moving a reference, the Court hereby expressly rejects any such interpretation as lacking support in the record.”</p>
6	<p>“a session associated with a transport layer protocol that is executed to convert one or more packets in a transport layer format into a different format”</p> <p>'683 Patent: Claim 10</p>	<p>Plain and ordinary meaning. No construction necessary.</p> <p>Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”</p>	<p>“convert the outermost header structure of the packet(s) from a transport layer protocol header to another type of header structure”</p> <p>The Court also concluded the following: “[t]o whatever extent Plaintiff contends that the terms ‘convert one or more packets having a TCP format into a different format,’ ‘convert one or more packets in a transport layer format into a different format,’ and ‘convert packets of the different format into another format’ encompass merely moving a reference, the Court hereby expressly rejects any such</p>

#	Plaintiff's Claim Term	Plaintiff's Proposed Construction	Sandvine Court Construction and Stipulation
			interpretation as lacking support in the record.”
7A	<p>“a second routine that is used to execute a second, different protocol to convert packets of the different format into another format”</p> <p>’683 Patent: Claim 2</p> <p>’790 Patent: Claim 3</p>	<p>Plain and ordinary meaning.</p> <p>No construction necessary.</p> <p>Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”</p>	<p>“operate on packets whose outermost header is a [second/third], different protocol header”</p>
7B	<p>“a [third] routine that is used to execute a [third], different [application layer] protocol to further convert the packets”</p> <p>’683 Patent: Claim 2</p> <p>’790 Patent: Claim 4</p>	<p>Plain and ordinary meaning.</p> <p>No construction necessary.</p> <p>Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”</p>	<p>“operate on packets whose outermost header is a [second/third], different protocol header”</p>
11	<p>“a routine that is used to execute a Transmission Control Protocol (TCP) to process packets having a TCP format”</p> <p>“a routine that is used to execute TCP to process at least one of the subsequent packets having a TCP format”</p> <p>’104 Patent: Claims 1, 10, 16</p>	<p>Plain and ordinary meaning.</p> <p>No construction necessary.</p> <p>Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”</p>	<p>“operate on one or more packets whose outermost header is a TCP header”</p>
12	<p>“a second routine that is used to execute a second protocol to process packets having a format other than the TCP format, wherein the second protocol is an application-level protocol”</p> <p>’104 Patent: Claim 3</p>	<p>Plain and ordinary meaning.</p> <p>No construction necessary.</p> <p>Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”</p>	<p>“execute a second protocol to operate on packets whose outermost header is other than a TCP header, wherein the second protocol is an application-level protocol”</p>

B. “Convert One or More Packets Having a TCP Format Into a Different Format” and similar terms

“convert one or more packets having a TCP format into a different format” “convert at least one or more of the packets of the message into a different format” “convert packets having a TCP format into a different format” ⁸	
Implicit’s Proposed Construction	Defendants’ Proposed Construction
Plain and ordinary meaning. No construction necessary. Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”	“convert the outermost header structure of the packet(s) from TCP to another type of header structure”

“convert one or more packets in a transport layer format into a different format” ⁹	
Implicit’s Proposed Construction	Defendants’ Proposed Construction
Plain and ordinary meaning. No construction necessary. Alternatively, “a session corresponding to a software routine for processing the packet from the transport layer to another layer in the protocol stack”	“convert the outermost header structure of the packet(s) from TCP to another type of header structure”

“convert packets of the different format into another format” ¹⁰	
Implicit’s Proposed Construction	Defendants’ Proposed Construction
Plain and ordinary meaning. No construction necessary. Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”	“convert each packet’s outermost header structure from the different protocol header into another type of header structure”

⁸ ’683 Patent: Claim 1; ’790 Patent: Claims 1, 8, 15; ’780 Patent: Claims 1, 16; ’839 Patent: Claim 1; ’378 Patent: Claims 1, 16.

⁹ ’683 Patent, Claim 10

¹⁰ ’683 Patent, Claim 2; ’790 Patent, Claim 3

1. Outermost Header

The Parties' disagreement as to the construction of these terms centers on the characterization of the format of the packet as the "outermost header structure." Defendants' constructions are the same as those the Court issued in *NetScout*, which the Court should adopt in this case as well. *NetScout* at 29. The intrinsic record plainly demonstrates that the "format" of a packet is its "outermost header structure." This is made clear in the disclosures in the specification (*see, e.g.*, '683 Patent at Abstract; Fig. 4, 1:24-44; 1:59-66; 3:62-67; 5:6-57; 6:56-50; 11:39-33; 14:4-17), as well as statements by Implicit in the '163 *inter partes* Reexam (*see* Ex. 2). In the latter, Implicit explained that the format of a packet *is defined* by its header structure: "Whether a packet is an IP format is determined by the structure of its header." Ex. 2 at 12. Implicit reinforced this principle by asserting that "[t]he format of the IP packet is not changed unless **the actual structure of the header itself** is changed from that shown in Figure 2 to a different type of header (*e.g.*, a TCP header)." *Id.* at 24 (emphasis added); *see id.*, at 12 (Fig. 2).

The Asserted Patents resulted from continuations of the '163 Patent, and these statements as to the meaning of "format," which is a term used extensively in the shared specification, are probative. *See, e.g., Alloc, Inc. v. Int'l Trade Comm'n*, 342 F.3d 1361, 1372 (Fed. Cir. 2003).

This Court should reject Implicit's attempt to rely on "expert" evidence to change these constructions at this late stage of Implicit's litigation campaign. Indeed, Implicit may not rely upon its expert at all for the reasons set forth in Juniper's motion to disqualify Dr. Almeroth due to an intractable conflict of interest. Dkt. 177.

Moreover, even if considered, Implicit's expert's declaration fails to provide any meaningful support for Implicit's proposed constructions. The declaration purports to describe certain kinds of packet processing but fails to show the claims encompass "conversion processing . . . through the use of pointers, using copies of the packets, other data structures, or other ways."

Dkt. 208 at 18. Implicit points to the language of claims 16, 20, and 24 in the '683 Patent as supporting its overbroad interpretation of the claims, and this Court properly rejected similar arguments in *NetScout*. *NetScout* at 25, 28. Expert testimony is the least useful form of extrinsic evidence, and is entitled to no weight where it is at odds with the intrinsic record, including the specific statements in the prosecution history. *Southwall Techs., Inc. v. Cardinal IG Co.*, 54 F.3d 1570, 1578 (Fed. Cir. 1995). Finally, as stated above, these terms were construed by the Court in *NetScout*, and Implicit is estopped from arguing for a different construction in this case.

2. Pointer Defining Outermost Header

Implicit's next attempt to re-write the Court's prior construction is to change the construction of "outermost header" to include a pointer "to a header within the packet." Dkt. 208 at 23. This is precisely the issue it lost previously. Implicit claims the declaration of Dr. Ng, submitted during the '857 Reexam, supports its argument. *Id.* But Dr. Ng opined—consistent with the Court's prior constructions—that "only the structure of the outermost header determines whether the packet is an IPv4 packet or whether it employs some other protocol." Dkt. 208-9 (Ng Decl.) at ¶ 6. Also, Implicit stated in *F5 Networks II* that "[e]ach message packet can include different layers in different data formats" and, for example, "software routines associated with an Ethernet protocol will process the packet first, as the outermost layer of the packet is in an Ethernet format." Ex. 3 at 4–5.

Judge Gilstrap considered a similar issue in Implicit's case against Palo Alto Networks, in which the Court rejected Implicit's proposal to construe "removing [an/the resulting] outermost header" in Claim 24 of the '683 Patent as "advancing the reference past the header information" or "either (1) stripping off or deleting a header; or (2) advancing a pointer past the header information." *PAN*, Dkt. 101 at 24. Like here, "Plaintiff replies that 'PAN ignores the realities of computer programming in order to arrive at a result that is inconsistent with the way in which

headers and data are passed between layers in any conventional networking stack.’” *Id.* at 23-24. Looking to Claims 16 and 20, the Court held, “[b]ecause Claim 20 recites ‘removing’ with reference to ‘convert[ing]’ from an input format to an output format (as recited in Claim 16), the ‘convert[ing]’ in Claim 20 can be fairly read as implying that the ‘removing’ involves modifying the packets rather than merely moving a reference.” *Id.* at 26. Thus, the Court “expressly reject[ed] Plaintiff’s proposed interpretation, in particular as to encompassing moving a ‘reference’ or ‘pointer.’” *Id.* at 28. Likewise, the claim terms at issue here involve converting one format “into a different format,” which is more than merely moving a reference or pointer, and demonstrates the necessity of limiting the format of the packet to the “outermost header structure.”

Indeed, the patentee evidently understood that the relevant “format” of a packet is determined by its outermost header. *NetScout* at 26. At best, Implicit shows only that pointers were well known when the above statements were made and that Implicit understood pointers existed, but did *not* include them within the claims. Instead, the claims require more than merely pointers to a header within the packet. For example, the specification of the ’683 Patent generally explains a concept of a “reference” or pointer; however, the claims do not import this concept. While it may be possible for a computer to read different portions of a packet by accessing memory locations, that is simply not what the claims recite. Thus, the outermost header determines the format, as the Defendants’ proposed constructions and the Court’s past constructions reflect, neither of which include a pointer to a header.

3. Moving a Reference

Implicit also argues again that these claim terms encompass merely moving a reference—which is another way of reviving its rejected a “pointer” argument. The Court already foreclosed Implicit’s interpretation as lacking support in the record:

This disclosure of “advanc[ing] the reference past the header information for the protocol so that the reference is positioned at the next header” [in the specification of the ’683 Patent] suggests that a packet could be handled in a manner that is at least somewhat independent of its outermost header. Yet, this disclosure relates to an operation that “may perform no conversion of the message.” [’683 Patent at 14:4–16.] Plaintiff has failed to demonstrate that anything in this disclosure is inconsistent with the patentee’s understanding, as apparent in the above-discussed evidence, that the format of a packet is determined by its outermost header.

Netscout at 27. Implicit claims that its additional citations to the intrinsic record somehow require the Court to clarify that a “system can advance a reference as part of converting a packet from one format to another format (*e.g.*, converting from a TCP format to an application format).” Dkt. 208 at 27. Implicit’s pointer argument, which it is collaterally estopped from making in the first place, should be rejected once again.

First, the prosecution history of the ’683 Patent confirms that “converting” requires modifying headers—not just moving a reference:

Assuming *arguendo* that Decasper’s plugins or flows correspond to the “sequence of routines” of claim 26 (which Applicant does not concede), Decasper does not teach or suggest that any of the plugins operates on “packets having a TCP *format*” let alone “*convert[ing]*” such packets “into a different *format*,” as recited in that claim. Rather, as discussed at length above, Decasper’s packet classification scheme relies on *IP headers remaining with packets* throughout the IP core.

See Ex. 1 at 19 (emphasis added). While it may be possible for a computer to read different portions of a packet by accessing memory locations, that is not what the claim recites. The claims plainly require “converting” the outermost header. Had Implicit intended to claim advancing references past headers, it could have done so. *See, e.g.*, ’104 Patent, claim 11 (“the created one or more data structures **include a reference** to the key value and to a queue for storing packets of the message for processing.”).

C. “[Execute] [Perform] a Transmission Control Protocol” and Related Terms

“execute a Transmission Control Protocol” “perform a Transmission Control Protocol” ¹¹	
Implicit’s Proposed Construction	Defendants’ Proposed Construction
Plain and ordinary meaning. No construction necessary. Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”	“operate on one or more packets whose outermost header is a TCP header”

“executable to perform a Transmission Control Protocol (TCP)” ¹²	
Implicit’s Proposed Construction	Defendants’ Proposed Construction
Plain and ordinary meaning. No construction necessary. Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”	“operable on one or more packets whose outermost header is a TCP header”

“execute a [second/different] [third/different] protocol” ¹³	
Implicit’s Proposed Construction	Defendants’ Proposed Construction
Plain and ordinary meaning. No construction necessary. Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”	“operate on packets whose outermost header is a [second/third], different protocol header”

¹¹ ’683 Patent, Claim 1; ’790 Patent, Claims 1, 15; ’104 Patent, Claims 1, 16; ’780 Patent, Claims 1, 16; ’839 Patent, Claim 1; ’378 Patent, Claims 1, 16.

¹² ’790 Patent, Claim 8.

¹³ ’683 Patent, Claim 2; ’790 Patent, Claim 3.

“execute a Transmission Control Protocol (TCP) to process packets having a TCP format”; “execute TCP to process at least one of the subsequent packets having a TCP format” ¹⁴	
Implicit’s Proposed Construction	Defendants’ Proposed Construction
Plain and ordinary meaning. No construction necessary.	“operate on one or more packets whose outermost header is a TCP header”
Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”	

“execute a second protocol to process packets having a format other than the TCP format, wherein the second protocol is an application-level protocol” ¹⁵	
Implicit’s Proposed Construction	Defendants’ Proposed Construction
Plain and ordinary meaning. No construction necessary.	“execute a second protocol to operate on packets whose outermost header is other than a TCP header, wherein the second protocol is an application-level protocol”
Alternatively, “a software routine for processing the packet from the TCP layer to another layer in the protocol stack”	

These claim terms and the Court’s clarification of what the terms did not mean—i.e., rejection of the argument Implicit rehashes in this case—were squarely addressed in *NetScout*, and were included in the Sandvine Stipulation. As a result, Implicit is collaterally estopped from arguing for different constructions. Moreover, as the Court previously noted, the dispute as to the construction of these claims “presents substantially the same dispute” as presented above in Section VII.B (*Netscout* at 33), and again, the intrinsic record supports the constructions. *See* Ex. 1 at 19 n.4 (“executes the TCP protocol (i.e., operates on a packet whose outermost header is a TCP header)”).)

¹⁴ ’104 Patent, Claims 1, 10, 16.

¹⁵ ’104 Patent, Claim 3.

VIII. “[S]ession Associated With A [Transport Layer/Different] Protocol” and similar terms

“session associated with a [transport layer/different] protocol”; “another session associated with a different protocol that is executed, wherein the different protocol corresponds to the different format” ¹⁶	
Implicit’s Proposed Construction	Defendants’ Proposed Construction
Plain and ordinary meaning. No construction necessary.	“information that is specific to a software routine for executing a specific protocol, that can be used for all packets of the message, and that is not information related to an overall path”

“TCP session associated with [the received] one or more [received] packets” ¹⁷ “a single TCP session” ¹⁸	
Implicit’s Proposed Construction	Defendants’ Proposed Construction
Plain and ordinary meaning. No construction necessary.	“information that is specific to a software routine for executing a TCP, that can be used for all packets of the message, and that is not information related to an overall path”

“sessions corresponding to [various/respective] ones of the sequence of [two or more] routines” ¹⁹	
Implicit’s Proposed Construction	Defendants’ Proposed Construction
Plain and ordinary meaning. No construction necessary.	“information that is specific to a software routine for executing a specific protocol that can be used for all packets of the message, and that is not information related to an overall path”

¹⁶ ’683 Patent, Claim 10.

¹⁷ ’790 Patent, Claims 6, 11, 20.

¹⁸ ’780 Patent, Claim 13.

¹⁹ ’790 Patent, Claims 5, 12, 17; ’104 Patent, Claim 5, 12.

In *NetScout*, this Court construed the term “state information” to mean “information that is specific to a software routine for a specific message, that can be used for all packets of the message, and that is not information related to an overall path.” *NetScout* at 19. The Asserted Patents claim and describe “state information” and “sessions” as interrelated concepts enabling the system to process all packets of the message efficiently. For example, the specification describes “state information as an instance or session of the conversion routine” and explains that “the session includes the protocol and state information associated with that instance of the protocol.” ’683 Patent at 3:1-9, 5:44-45; *see also id.* at 3:60-62 (“sessions are identified so that each packet is associated with the appropriate state information.”) Defendants submit that the Court’s prior construction for the term “state information” (agreed to by the Parties here) should apply for—as an instance of, or included within—the equivalent term “sessions” as used in the claims.

Defendants’ constructions find ample support in the intrinsic record tying sessions of the conversion routines to state information. The claims equate “sessions” with “state information” by claiming that “*sessions specify state information* for one or more of the sequence of routines.” ’683 Patent, claims 5, 11 (emphasis added). The specification likewise equates “state information as an instance or session of the conversion routine,” as quoted in *NetScout*:

[S]ince the conversion routines may need to retain state information between the receipt of one packet of a message and the next packet of that message, *the conversion system maintains state information as an instance or session of the conversion routine*. The conversion system routes all packets for a message through the same session of each conversion routine so that the same state or instance information can be used by all packets of the message.

NetScout at 17 (quoting ’683 Patent at 3:1-9) (emphasis added). According to Implicit, once the state information-sessions are configured within the system, the invention “routes all packets for a message through the *same session of each conversion routine*” to ensure “the *same state information* can be used by *all packets* in the message.” ’683 Patent at 3:5-9 (emphasis added).

Implicit relied on this disclosure in the '163 *inter partes* Reexam, arguing that the claimed “message-based technology fundamentally differs from a packet-based technology” found in the prior art routers. Ex. 2 at 2. Implicit again equated a “session” with “state information,” explaining that within the conversion system:

a “session” is created to ensure that all packets of a message are processed properly: the “conversion system routes ***all packets of a message*** through the same session of each conversion routine so that the same state or instance information can be used by ***all packets of the message***.”

Id. (quoting '683 Patent at 3:2-7) (emphasis in original); *see also NetScout* at 17. This Court found Implicit’s statements to disclaim “soft state” found in prior art routers from “hard state [] tied to the processing of entire messages” maintained in the claimed invention. *See NetScout* at 19 (citing Ex. 2 at 11); *see also id.* at 17-19 (citing Ex. 2 at 2, 10-11, 13-14, 26-28, 38-39). This disclaimer was deemed consistent with Implicit’s reliance on the specification’s disclosure of routing all packets through the *same session of each conversion routine* so that “the same state or instance information can be used by all packets of the message.” *Id.* at 19 (citing '683 Patent at 3:1-9); Ex. 2 at 2.

Implicit’s attempts to argue that “session” is unrelated to “state information” to avoid its disclaimer are thus belied by the claims themselves and Implicit’s many statements equating “session” and “state information” in the detailed description and in reexamination. Defendants’ constructions should be adopted.

IX. Conclusion

For the foregoing reasons, Defendants respectfully request that the Court adopt Defendants’ proposed constructions.

Dated: February 18, 2020

By: /s/ David McPhie

IRELL & MANELLA LLP

David McPhie (*pro hac vice*)
(CA State Bar No. 231520)
Ingrid Petersen (*pro hac vice*)
(CA State Bar No. 313927)
840 Newport Center Drive, Suite 400
Newport Beach, CA 92660
Telephone: 949-760-0991
Facsimile: 949-760-5200
dmcphie@irell.com
ipetersen@irell.com

Jonathan S. Kagan (*pro hac vice*)
(CA State Bar No. 166039)
1800 Avenue of the Stars, Suite 900
Los Angeles, CA 90067-4276
Telephone: 310-277-1010
Facsimile: 310-203-7199
jkagan@irell.com

GILLAM & SMITH LLP

Melissa Richards Smith
(Texas Bar No. 24001351)
303 South Washington Avenue
Marshall, TX 75670
Telephone: 903-934-8450
Facsimile: 903-934-9257
melissa@gillamsmithlaw.com

Attorneys for Defendant Juniper Networks, Inc.

Dated: February 18, 2020

By: /s/ Michael J. Sacksteder

FENWICK & WEST LLP

Michael J. Sacksteder (Lead Attorney)
CA Bar No. 191605 (Admitted E.D. Tex.)
Christopher L. Larson
CA Bar No. 308247 (Admitted E.D. Tex.)
Jessica L. Benzler
CA Bar No. 306164 (Admitted E.D. Tex.)
FENWICK & WEST LLP
555 California Street, 12th Floor
San Francisco, CA 94104

Telephone: 415-875-2300
Facsimile: 415-281-1350
msacksteder@fenwick.com
clarson@fenwick.com
jbenzler@fenwick.com

Geoffrey R. Miller
(Texas State Bar No. 24094847)
801 California Street
Mountain View, CA 94041
Telephone: 650-335-7943
Facsimile: 650-938-5200
gmiller@fenwick.com

Attorneys for Defendant Imperva, Inc.

Dated: February 18, 2020

By: /s/ Matthew C. Gaudet

THE DACUS FIRM, P.C.

Deron R. Dacus
State Bar No. 00790553
821 ESE Loop 323, Suite 430
Tyler, TX 75701
Phone: 903-705-1117
Facsimile: 903-581-2543
ddacus@dacusfirm.com

DUANE MORRIS LLP

Matthew C. Gaudet (GA SBN 287789)
Admitted E.D. Tex.
David C. Dotson (GA SBN 138040)
Admitted E.D. Tex.
John R. Gibson (GA SBN 454507)
Admitted E.D. Tex.
Alice E. Snedeker
Admitted E.D. Tex.
1075 Peachtree NE, Suite 2000
Atlanta, GA 30309
Telephone: 404-253-6900
Facsimile: 404-253-6901
mcgaudet@duanemorris.com
dcdotson@duanemorris.com
jrgibson@duanemorris.com
aesnedeker@duanemorris.com

Christopher J. Tyson (VA SBN 81553)
Admitted E.D. Tex.
505 9th Street, N.W., Suite 1000
Washington, DC 20004-2166
Telephone: 202-776-7800
Facsimile: 202-776-7801
cjtyson@duanemorris.com

Counsel for Defendant Fortinet, Inc.

CERTIFICATE OF SERVICE

The undersigned certifies that the foregoing document was filed electronically on February 18, 2020, in compliance with Local Rule CV-5(a). As such, this document was served on all counsel who are deemed to have consented to electronic service. Local Rule CV-5(a)(3)(A).

/s/ Michael J. Sacksteder

Michael J. Sacksteder